
Exercice I - Crible d'Eratosthène.

Description de l'algorithme : pour trouver les nombres premiers compris entre 2 et N . On écrit dans un tableau les entiers de 2 à N . Tant qu'il reste des entiers non rayés ou non entourés on réalise les actions suivantes

- On entoure dans le tableau le plus petit entier.
- On raye les multiples de cet entier.

Les éléments entourés sont les nombres premiers.

Écrire un algorithme qui utilise cette méthode pour donner les nombres premiers jusqu'à un entier déterminé par l'utilisateur.

Exercice II - Dichotomie.

Partie mathématique. On considère l'application de \mathbb{R} dans \mathbb{R} définie par $f(x) = \cos(x) - x$.

1. Montrer qu'il existe une valeur x_0 qui annule f .
2. Cette valeur est-elle unique?
3. On définit les suites (u_n) et (v_n) par récurrence en posant $u_0 = 0$, $v_0 = \frac{\pi}{2}$ et :

Cas 1 : si $f(u_n)$ et $f\left(\frac{u_n+v_n}{2}\right)$ sont de même signe,

$$\begin{cases} u_{n+1} &= \frac{u_n+v_n}{2} \\ v_{n+1} &= v_n \end{cases}$$

Cas 2 : si $f(u_n)$ et $f\left(\frac{u_n+v_n}{2}\right)$ sont de signes différents,

$$\begin{cases} u_{n+1} &= u_n \\ v_{n+1} &= \frac{u_n+v_n}{2} \end{cases}$$

Montrer que pour tout n de \mathbb{N} , on a : $|u_n - v_n| \leq \frac{\pi}{2^{n+1}}$.

4. Montrer que les suites (u_n) et (v_n) sont adjacentes.
5. En déduire que (u_n) et (v_n) convergent vers x_0 .

Partie informatique. Écrire un algorithme qui permet de déterminer une valeur approchée à 10^{-12} de x_0 .

Mini-projet 1 (Reprise) : le mastermind

Le but est de créer un programme avec lequel on peut jouer au Mastermind. L'ordinateur choisit quatre couleurs différentes choisies dans la variable globale :

$$\text{Couleurs} = ['R', 'B', 'V', 'N', 'O', 'J']$$

et l'utilisateur doit les trouver en un minimum d'essai. A chaque essai, l'ordinateur donnera le nombre de pions de la bonne couleur et bien placés (BP) et le nombre de pions de la bonne couleur mais mal placés (MP). Exemple :

Pions cachés	<table border="1"><tr><td>R</td><td>V</td><td>B</td><td>N</td></tr></table>	R	V	B	N	
R	V	B	N			
Essai 1	<table border="1"><tr><td>V</td><td>J</td><td>O</td><td>N</td></tr></table>	V	J	O	N	1BP, 1MP
V	J	O	N			
Essai 2	<table border="1"><tr><td>R</td><td>B</td><td>V</td><td>N</td></tr></table>	R	B	V	N	2BP, 2MP
R	B	V	N			
Essai 3	<table border="1"><tr><td>B</td><td>R</td><td>V</td><td>N</td></tr></table>	B	R	V	N	1BP, 3MP
B	R	V	N			
Essai 4	<table border="1"><tr><td>R</td><td>V</td><td>B</td><td>N</td></tr></table>	R	V	B	N	4BP, 0MP
R	V	B	N			

1. Écrire une fonction *CouleurDifférente* qui reçoit une liste et qui vérifie que cette liste contient des éléments différents.
2. Écrire une fonction *Initialise* qui renvoie une liste contenant 4 couleurs différentes prises dans *Couleurs*. On vérifiera bien que ce choix est possible.

Note : On pourra se servir de la fonction *randint(a, b)* de la bibliothèque *random* qui renvoie un entier aléatoire entre *a* et *b* compris.

3. Écrire une fonction *Compare* qui à partir de deux listes indique le nombre de couleurs bien placées et de couleurs mal placées.
4. Écrire une fonction *DemandeCouleur* qui demande à l'utilisateur un choix de *NbDeCouleurs* couleurs et qui vérifie que les couleurs sont dans *Couleurs* et qu'elles sont bien différentes.
5. Écrire un programme permettant de jouer au Mastermind avec l'ordinateur.
6. Adapter le travail précédent pour autoriser à cacher plusieurs fois la même couleur. L'éditeur du jeu annonce "2401 combinaisons possibles". Expliquer le calcul menant à ce résultat.